

AKSHAT18FEB@GMAIL.COM

[GITLINK](#)

[LINKEDIN](#)

BRICK OVEN



SQL PROJECT

BRICK OVEN



NEXT >



ABOUT ME

Myself, Akshat Waghmare is a motivated Electronics and Telecommunication Engineering graduate with a growing expertise in data science, cloud computing, and quality assurance. Also has hands-on experience in MySQL. Showcased through a pizza sales project where created Database, Tables and solved complex SQL queries.

< BACK

NEXT >





Q1.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



```
select count(order_id) from orders;
```

Result Grid | Filter

| count(order_id) |
|-----------------|
| 21350 |

< BACK

NEXT >



Q2. IDENTIFY THE HIGHEST-PRICED PIZZA.

```
select name,price from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
order by price desc limit 1;
```

Result Grid | Filter Rows:

| | name | price |
|---|-----------------|-------|
| ▶ | The Greek Pizza | 35.95 |

< BACK

NEXT >



Q3. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

**SELECT**`ROUND(SUM(quantity * price),2) AS total_revenue`**FROM**`order_details`**JOIN**`pizzas ON pizzas.pizza_id = order_details.pizza_id;`

Result Grid | Filter

| | total_revenue |
|---|---------------|
| ▶ | 817860.05 |

< BACK

NEXT >



Q4. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.



```
select hour(order_time) as Order_Hourly, count(order_id) as order_count from orders  
group by Order_Hourly;
```

Result Grid | Filter Rows:

| | Order_Hourly | order_count |
|---|--------------|-------------|
| ▶ | 11 | 1231 |
| | 12 | 2520 |
| | 13 | 2455 |
| | 14 | 1472 |
| | 15 | 1468 |
| | 16 | 1920 |
| | 17 | 2336 |
| | 18 | 2399 |
| | 19 | 2009 |
| | 20 | 1642 |
| | 21 | 1198 |

< BACK

NEXT >



Q5. GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.



```
select round(avg(Quantity),2) from  
(select order_date , sum(quantity) as Quantity  
from order_details join orders  
on order_details.order_id = orders.order_id  
group by (order_date)) as data;
```

Result Grid | Filter Rows:

| | |
|---|------------------------|
| | round(avg(Quantity),2) |
| ▶ | 138.47 |

< BACK

NEXT >



Q6. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.



```
select category , round((sum(quantity * price) / (SELECT  
    ROUND(SUM(quantity * price),2) AS total_revenue  
FROM  
    order_details  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id))*100,2) as Revenue from  
pizza_types join pizzas on  
pizza_types.pizza_type_id = pizzas.pizza_type_id  
join order_details on  
order_details.pizza_id = pizzas.pizza_id  
group by category  
order by Revenue desc;
```

| Result Grid | | |
|-------------|----------|---------|
| | category | Revenue |
| ▶ | Classic | 26.91 |
| | Supreme | 25.46 |
| | Chicken | 23.96 |
| | Veggie | 23.68 |

< BACK

NEXT >



Q7. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```

select name,Revenue from
(select name,category, Revenue,
rank() over(partition by category order by Revenue desc) as rn
from
(select name,category, sum(quantity*price) as Revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by name,category) as A) as B
where rn <=3;
    
```

Result Grid | Filter Rows:

| | name | Revenue |
|---|------------------------------|------------------|
| ▶ | The Thai Chicken Pizza | 43434.25 |
| | The Barbecue Chicken Pizza | 42768 |
| | The California Chicken Pizza | 41409.5 |
| | The Classic Deluxe Pizza | 38180.5 |
| | The Hawaiian Pizza | 32273.25 |
| | The Pepperoni Pizza | 30161.75 |
| | The Spicy Italian Pizza | 34831.25 |
| | The Italian Supreme Pizza | 33476.75 |
| | The Sicilian Pizza | 30940.5 |
| | The Four Cheese Pizza | 32265.7000000065 |
| | The Mexicana Pizza | 26780.75 |



< BACK

NEXT >

[GITLINK](#)

[LINKEDIN](#)

BRICK OVEN



THANK YOU!

< BACK